A9

DECISION TREE GENERATING DEVICE

Patent number: JP9330224
Publication date: 1997-12-22

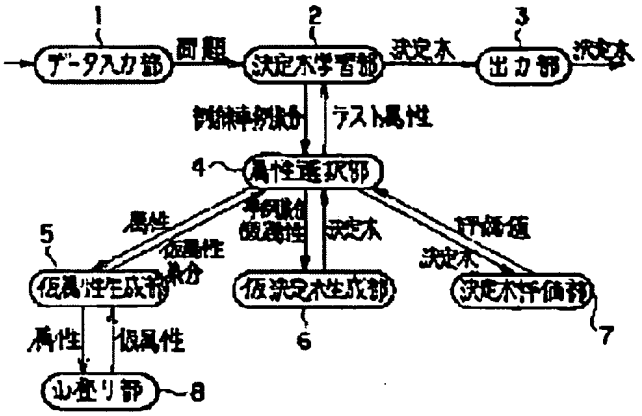
Inventor: YUGAMI NOBUHIRO; OOTA TADAKO
Applicant: FUJITSU LTD

Classification:
- international: G06F9/44
- european:
Application number: JP19960147497 19960610
Priority number(s):

Abstract of **JP9330224**

PROBLEM TO BE SOLVED: To automatically generate a decision tree of excellent classification precision at a small frequency of processing by evaluating a decision tree which can branch off when a decision tree is generated from a set of instances and generating the decision tree having the largest evaluated value.

SOLUTION: When judging that an instance set inputted and supplied from a data input part 1 or an instance set after decision meets end conditions, a decision tree learning part 2 generates leaf node to generate a decision tree as a result, which is outputted by an output part 3. When the end conditions are, not met, on the other hand, an attribute selection part 4 sends an instruction to a tentative attribute generation part 5 to generate an attribute which can branch off or its value in the object instance set, a tentative tree generation part 6 generates a tentative tree on the basis of the generated attribute or its value, and a decision tree evaluation part 7 calculates an evaluated value about the generated tentative decision tree and selects the attribute generating the tree having the largest evaluated value repeatedly. At the time of branching, unnecessary division is evaded and a down-hill method is employed.



(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平9－330224

(43)公開日　平成９年(1997)12月22日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　9/44	５５０		Ｇ０６Ｆ　9/44	５５０Ｎ	

審査請求　未請求　請求項の数３　ＯＬ　（全 12 頁）

(21)出願番号　特願平8－147497

(22)出願日　平成８年(1996)６月10日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　湯上　伸弘
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

(72)発明者　太田　唯子
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

(74)代理人　弁理士　岡田　守弘

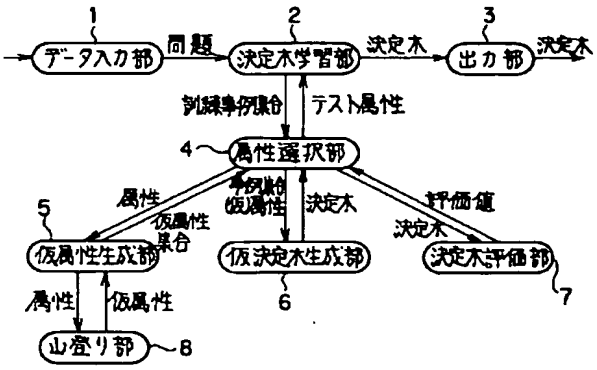
(54)【発明の名称】　決定木生成装置

(57)【要約】

【課題】　本発明は、事例集合の決定木を生成する決定木生成装置に関し、事例集合から決定木を生成するときに分岐可能な決定木を評価して評価値の最も高い決定木を生成することを繰り返すと共に、分岐するときに不要な分割を回避および山登り法を採用し、少ない処理回数で分類精度の良好な決定木を自動生成することを目的とする。

【解決手段】　与えられた事例集合あるいは分割後の事例集合について、終了条件を満たしている場合にリーフノードを生成して結果を出力する手段と、終了条件を満たしていない場合に、対象となる上記事例集合中に分割可能な属性あるいは属性の値で分岐する仮の決定木を生成する手段と、生成された仮の決定木についてそれぞれの評価値を計算して評価値の最も高い仮の決定木について、決定ノードおよび属性値をエッジで表現した決定木を生成することを繰り返す手段とを備えるように構成する。

本発明のシステム構成図



1

【特許請求の範囲】
【請求項１】事例集合から決定木を生成する決定木生成装置において、
与えられた事例集合あるいは分割後の事例集合について、終了条件を満たしている場合にリーフノードを生成して結果を出力する手段と、
上記終了条件を満たしていない場合に、対象となる上記事例集合中に分割可能な属性あるいは属性の値で分岐する仮の決定木を生成する手段と、
上記生成された仮の決定木についてそれぞれの評価値を計算して評価値の最も高い仮の決定木について、決定ノードおよび属性値をエッジで表現した決定木を生成することを繰り返す手段とを備えたことを特徴とする決定木生成装置。
【請求項２】上記分割可能な属性の値で分岐する際に、同じような結果が得られる属性の値をまとめて１つにする仮の属性で分岐することにより、不要な分岐を回避することを特徴とする請求項１記載の決定木生成装置。
【請求項３】上記分割可能な属性の値で分岐する際に、属性が取り得る値の任意の２つを１つにまとめて上記評価を行って最も評価の高い値のペアを１つにまとめる仮の属性を選ぶことを繰り返した後、全体の中で最も評価値の高い仮の属性の値で分岐し、属性が多値を取るときに処理回数を削減することを特徴とする請求項２記載の決定木生成装置。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、事例集合から決定木を生成する決定木生成装置に関するものであって、機械の故障診断や病気の診断などの分類問題における決定木を自動生成する決定木生成装置に関するものである。
【０００２】
【従来の技術】従来、機械の故障診断や病気の診断などの分類問題における決定木である例えば図１０に示す決定木を生成する場合、次のようにして行っていた。ここで、図１０に示す決定木は、図９に示す属性（Ｘ、Ｙ、Ｚなどの属性）とその値（Ｘの値が“０”あるいは“１”など）のペアの集合で表現された事例を分類する*

2

*ための規則を木構造で表現したものである。決定木は、ノードとノード間を結ぶエッジからなる。ノードは、決定ノードとリーフノードの２種類がある。決定ノードは属性が割り当てられており、リーフノードにはクラス（分類結果）が割り当てられている。決定木の根のノードをルートノードという。
【０００３】決定木を用いて事例を分類するためには、以下のような再帰アルゴリズムを用いる。
入力:事例
出力:クラス
ステップ１
ノードNをルートノードに初期化する。
【０００４】ステップ２
もしノードNがリーフノードであるならば、Nに割り当てられているクラスを出力して、終了する。
【０００５】ステップ3
Nに、分類したい事例の、Nに割り当てられている属性の値に対応するNの子ノード(Nと、属性の値に対応するエッジで結ばれているノード)を代入し、ステップ2へ進む。
【０００６】一般に、決定木の、ルートノードからリーフノードにいたるパスは、パス中の決定ノードの属性と、決定ノードから出ている値によって定義される条件の連言(決定ノードは、ひとつのパス中に複数存在する場合もある)を条件部とし、リーフノードのクラスを結論とするルールと等価である。
【０００７】この際、以下で説明するインフォメーションゲインと呼ばれる評価基準を用いてテスト属性を決定することを特徴とする決定木生成方式(以下ID3と呼ぶ)がある。インフォメーションゲインは、与えられた訓練事例集合を属性の値で分割することによる分割前後の事例集合(分割後は集合の集合)中のクラスの偏りを表わすエントロピーの変化量である。
【０００８】以下で、Sを事例集合、N(S)をSに含まれる事例の数、Ni(S)をSに含まれる事例のなかでクラスがｉであるものの数、Sajを、Sに含まれる事例のなかで、属性aの値がjであるものの集合とする。このとき、事例集合SにおけるエントロピーH(S)は

$$H(S)=-\sum_{}\frac{Ni(S)}{N(S)}\log\frac{Ni(S)}{N(S)} \qquad \text{(式1)}$$

ここで対数の底は2である。インフォメーションゲインは分割前後のエントロピーの差であるが、分割後は複数の集合があるので、インフォメーションゲインは以下の※

※ようになる。
【０００９】

$$InfoGain(S,a)=H(S)-\sum_{j\in domain(a)}\frac{N(Saj)}{N(s)}H(Saj) \qquad \text{(式2)}$$

ID3は、この評価値を用いてテスト属性を選択して、事例集合を分割していき、集合に含まれる全ての事例が同じクラスに属するか、あるいは集合を分割できる属性がなくなったら、リーフノードを生成し、事例集合のなかで多数派をしめるクラスを割り当てるようにしている。
【００１０】

3

【発明が解決しようとする課題】上述したID3などの従来の決定木生成方式では、決定ノードにおけるテスト属性を決める際に、訓練事例集合を属性の値で分割した結果できる複数の訓練例集合内のクラスの分布のばらつきによって、属性を評価する。その属性で分割したことによって、分割後の各集合内で、ある特定のクラスに属する訓練例の割合が高くなれば、その属性による分割は有効であると判定される。この方法は、属性とクラスが直*

4

*接関係がある場合には、うまく働くが、属性とクラス間の関係が複雑な場合にはうまく働かない。これを以下のような簡単な例を用いて説明する。

【００１１】・クラスの定義(決定木生成時には未知)
XとYのxor、すなわち(X=1 and Y=0) or (X=0 and Y=1)ならば正、それ以外ならば負とする。

【００１２】

・属性と属性値の集合

属性名	属性値の集合
X	0, 1
Y	0, 1
Z	0, 1

(テーブル1)

・訓練事例(以下の8個)

事例番号	属性値 X	属性値 Y	属性値 Z	クラス
1	0	0	0	負
2	0	0	1	負
3	0	1	0	正
4	0	1	1	正
5	1	0	0	正
6	1	0	1	正
7	1	1	0	負
8	1	1	1	負

(テーブル2)

この問題におけるクラスの定義(分類規則)は、「もしXの値とYの値が等しければクラスは負、そうでないならば正」である。すなわち、属性XとYはクラスの定義に関係しているが、属性Zは無関係である。よって、決定木中にあらわれるテスト属性としてはXとYは望ましいが、Zがあらわれる決定木は望ましくないことになる。

【００１３】このような問題に対して、ID3がどのように働くかを具体的に判りやすく説明する。ここで、上述した(テーブル2)に示すように、値として0または1をとる3個の属性X、Y、Zが存在し、事例のクラスが属性Xと属性Yの値の排他的論理和(xor)で決められているとして、ID3(従来技術)について以下説明(本願発明については実施例の中で説明)する。2つの値をとり得る属性が3個あるから上述した(テーブル2)に示すように、可能な事例は8個である。ここでは、(テーブル2)の8個の事例のうち、事例2、3、4、5、6、8の6個の事例が訓練事例として与えられたとする。

【００１４】ここで、ID3(従来技術)の手順を一般的に述べると以下の通りである。

・入力:訓練事例の集合TS
・出力:TSに対する決定木

ステップ1:もしTS中の全ての訓練事例が同じクラスCに属するならば、クラスCのリーフノードを生成し、それを出力する。

【００１５】ステップ2:属性のなかで、その値によって訓練事例集合TSに含まれる事例を分割可能な属性(TSに含まれる事例のペアのなかで、その属性の値が異なるものが存在する)の集合をAとする。もしAが空集合なら、TSに含まれる事例のなかで最も多い事例が属するクラスのリーフノードを生成し、それを出力する。

【００１６】ステップ3:属性集合Aに含まれる全ての属性について、インフォメーションゲインを計算し、それを最大とする属性aを選択する(インフォメーションゲインが最大となる属性が複数存在する場合は、それらのなかからランダムに選択する)。

【００１７】ステップ4:訓練事例集合TSを属性aの値で分割する。TSに含まれる事例のなかで属性aの値がvである事例の集合をTSvとする。

【００１８】ステップ5:全てのvについて、もしTSvが空集合でないなら、ID3(TSv)を実行し、TSvに対する　決定木DTvを生成する。

【００１９】ステップ6:テスト属性がaであり、a=v(vは属性aの値)に対応する子ノードがDTvのルートノードである決定ノードを生成する。生成きれた決定ノードがルートノードである決定木を生成し、それを出力する。

【００２０】ここで、インフォメーションゲインの定義については、(式2)を参照して下さい。次に、一般的な上記ステップ1ないしステップ6について、(テーブル2)の具体例のうち、ID3(従来技術)では訓練事例からなる集合({2,3,4,5,6,8})を引数として呼び出すことにより、決定木が生成される。 このとき、どのように動作するかを以下に示す。ここで、1:、2:等は、上記ステッ

5

プ1、ステップ2等に対応することを表わす。

【００２１】ID3({2,3,4,5,6,8}):

1:訓練事例集合TS={2,3,4,S,6,8}の中にはクラスが正の事例が4個、負の事例が2 個含まれるので、ステップ1はパスされる。

【００２２】2:TS中に含まれている事例のうち、事例3と事例6とでは属性Xの値が異なるので、XはTSを分割可能である。同様に、属性Y,ZもTSを分割可能であるので、A={X,Y,Z}となる。

【００２３】3:Aに含まれる属性について．インフォメーションゲインを計算すると、以下のようになるので、最適な属性aとして、Zが選択される。

属性	インフォメーションゲイン
X	0
Y	0
Z	0.251629

4:TSを属性Zの値で分割する。

【００２４】TS0={3,5}　　(Zの値が0)

TS1={2,4,6,8}　　(Zの値が1)

5:TS0、TS1に対する決定木DT0、DT1をID3を使って生成する。

【００２５】DT0=ID3({3,5})1:事例3および5のクラスはどちらも正であるので、クラスが正であるリーフノードを生成し、それを出力してID3({3.5})を終了する。

【００２６】DT1=ID3({2,4,6,8})

1:事例2と事例8のクラスは負、4と6は正であるので、ステップ1はパス。

【００２７】2:TS={2,4,6,8}に含まれる4個の中例の属性Zの値は全て1であるので、ZはTSを分割可能でない。事例2と事例6は属性Xの値が異なるので、XはTSを分割可能である。事例2と事例4は属性Yの値が異なるので、属性YもTSを分割可能である。よって、A={X,Y}となる。

【００２８】3:Aに含まれる属性について、訓練事例集合{2,4,6,8}に対するインフォメーションゲインを計算すると、以下のようになる。

属性	インフォメーションゲイン
X	0
Y	0

XとYのインフォメーションゲインが等しいので、最適な属性aとして、XまたはYをランダムに選択する。ここではXが選択されたとする。

【００２９】4:訓練事例集合TS={2,4,6,8}を、Xの値で分割する。

TS0={2,4}　　(Xの値が0)

TS1={6,8}　　(Xの値が1)

5:TS0,Ts1に対する決定木DT0,DT1をID3を使って生成する。

【００３０】DT0=ID3({2,4})

1:事例2のクラスは負、事例4は正であるので、ステップ1はパス。

6

【００３１】2:事例2と4はどちらも属性Xの値が0、属性Zの値が1であるので、TS={2,4}を分割可能な属性はYのみ。すなわち、A={Y}となる。

3:属性Yのインフォメーションゲインを計算する。結果は1。Aには属性Yしか含まれないので、最適な属性aとしては、属性Yが選択される。

【００３２】4:TS={2,4}を属性Yの値で分割する。

TS0={2}　　(Yの値が0)

TS1={4}　　(Yの値が1)

5:TS0、TS1に対する決定木DT0,DT1をID3を使って生成する。

【００３３】DT0=ID3({2})

1:事例2はクラス負に属するので、クラス負のリーフノードを生成し、それを出力して、ID3({2})を終了。

【００３４】DT1=ID3({4})

1:事例4はクラス正に属するので、クラス正のリーフノードを生成し、それを出力して、ID3({4})を終了。

【００３５】6:テスト属性がYであって、Y=0に相当する子ノードがDT0のルートノード、Y=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。この決定ノードがルートノードとなる決定木を出力し、ID3({2,4})を終了する。

【００３６】DT0=ID3({6,8})

1:事例6のクラスは正、事例8は負であるので、ステップ1はパス。

【００３７】2:事例6と8はどちらも属性Xの値が1、属性Zの値が1であるので、TS={6,8}を分割可能な属性はYのみ。すなわち、A={Y}となる。

3:属性Yのインフォメーションゲインを計算する。結果は1。Aには属性Yしか含まれないので、最適な属性aとしては、属性Yが選択される。

【００３８】4:TS={6,8}を属性Yの値で分割する。

TS0={6}　　(Yの値が0)

TS1={7}　　(Yの値が1)

5:TS0、TS1に対する決定木DT0,DT1をID3を使って生成する。

【００３９】DT0=ID3({6})

1:事例6はクラス正に属するので、クラス正のリーフノードを生成し、それを出力して、ID3({6})を終了。

【００４０】DT1=ID3({8})

1:事例8はクラス負に属するので、クラス負のリーフノードを生成し、それを出力して、ID3({8})を終了。

【００４１】6:テスト属性がYであって、Y=0に相当する子ノードがDT0のルートノード、Y=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。この決定ノードがルートノードとなる決定木を出力し、ID3({6,8})を終了する。

【００４２】6:テスト属性がXであって、X=0に相当する子ノードがDT0のルートノード、X=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。こ

の決定ノードがルートノードとなる決定木を出力し、ID3({2,4,6,8})を終了する。

【００４３】6:テスト属性がZであって、Z=0に相当する子ノードがDT0のルートノード、Z=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。この決定ノードがルートノードとなる決定木を出力し、ID3({2,3,4,5,6,8})を終了する。

【００４４】以上の結果として生成される決定木は図１１のようになる。この決定木は求めたいクラスの定義、すなわち、XとYの排他的論理和を正しく表わしてはいない。例えば、訓練事例として与えられていない事例、事例1と事例7はどちらも負に属するが、ID3で生成した図１１の決定木では正に分類されてしまう。このように、従来の上記方法などは、属性とクラス間の関係が明らかでない場合に、クラスを定義する属性を判別できず、その結果、生成される決定木の分類精度が悪くなってしまうという問題があった。

【００４５】本発明は、これらの問題を解決するため、事例集合から決定木を生成するときに分岐可能な属性について仮の決定木を生成し、その決定木を評価して評価値の最も高い決定木を生成した属性で分岐することを繰り返すと共に、分岐するときに不要な分割を回避および山登り法を採用し、少ない処理回数で分類精度の良好な決定木を自動生成することを目的としている。

【００４６】

【課題を解決するための手段】図１を参照して課題を解決するための手段を説明する。図１において、データ入力部１は、事例の問題を入力するものである。

【００４７】決定木学習部２は、入力された事例をもとに決定木を決定するものである。出力部３は、決定された決定木を出力するものである。仮属性生成部５は、対象となる事例について分割可能な仮の属性を生成するものである。

【００４８】仮決定木生成部６は、仮の決定木を生成するものである。決定木評価部７は、仮の決定木の評価を行うものである。山登り部８は、属性が多値を取るときに山登り法によって処理回数を削減するものである。

【００４９】次に、動作を説明する。データ入力部１から入力されて与えられた事例集合あるいは分割後の事例集合について、決定木学習部２が終了条件を満たしていると判断した場合にリーフノードを生成して結果の決定木を生成し、出力部３が出力するようにしている。一方、終了条件を満たしていない場合には、属性選択部４が仮属性生成部５に指示して対象となる事例集合中に分割可能な属性あるいは属性の値を生成させ、仮決定木生成部６に生成した属性あるいは属性の値をもとに仮の決定木を生成し、決定木評価部７に生成した仮の決定木について評価値を計算させ、最も評価値の高い決定木を生成した属性を選択することを繰り返すようにしている。

【００５０】この際、分割可能な属性の値で分岐する際に、同じような結果が得られる属性の値をまとめて１つにし、不要な分岐を回避するようにしている。また、山登り部８が分割可能な属性の値で分岐する際に、属性が取り得る値の任意の２つを１つにまとめて評価を行って最も評価の高い属性を選ぶことを繰り返した後、全体の中で最も評価値の高い属性の値で分岐し、属性が多値を取るときに処理回数を削減するようにしている。

【００５１】従って、事例集合から決定木を生成するときに分岐可能な属性について仮の決定木を生成し、その決定木を評価して評価値の最も高い決定木を生成することを繰り返すと共に、分岐するときに不要な分割を回避および山登り法を採用することにより、少ない処理回数で分類精度の良好な決定木を自動生成することが可能となる。

【００５２】

【発明の実施の形態】次に、図２から図９を参照して本発明の具体的な動作を順次詳細に説明する。図２ないし図７は本発明の具体例を示し、図９は事例を示す。

【００５３】ここで、図９の事例中の事例番号の２、３、４、５、６、８の６個の事例集合({2,3,4,5,6,8})を引数として下記の本発明のPDTを呼び出すことにより、決定木が自動生成される。ここで、1:、2:などは、PDT中のステップ1、ステップ2を表す。

【００５４】・PDT(TS)(本発明の以下のステップ1ないしステップ7を実行する手順名であって、PDTと呼ぶ)
入力:訓練事例の集合TS。

【００５５】出力:TSに対する決定木。
ステップ1:もしTS中の全ての訓練事例が同じクラスCに属するならば、クラスCのリーフノードを生成し、それを出力する。

【００５６】ステップ2:属性のなかで、その値によって訓練事例集合TSに含まれる事例を分割可能な属性(TSに含まれる事例のペアのなかで、その属性の値が異なるものが存在する)の集合をAとする。もしAが空集合なら、TSに含まれる事例のなかで最も多い事例が属するクラスのリーフノードを生成し、それを出力する。

【００５７】ステップ3:属性集合Aに含まれる全ての属性bについて、以下のようにして決定木DTbを生成する。
ステップ3-1:TSを属性bの値で分割する。TSに含まれる事例のなかで属性bの値がvである事例の集合をTSb=Vとする。

【００５８】ステップ3-2:各vについて、ID3(TSb=V)を実行する。その出力をDTb=vとする。
ステップ3-3:テスト属性がbであり、b=v(W;属性bの値)に対応する子ノードがDTb=Vのルートノードである決定ノードを生成し、その決定ノードがルートノードである決定木をDTbとする。

【００５９】ステップ4:属性集合Aに含まれる属性bを、DTbのラプラスエラーレートにより評価し、最適な属性を選択する(最適なものが複数ある場合は、それらのな

かからランダムに選択する)。選択された属性をaとする。

【００６０】ステップ5:訓練事例集合TSを属性aの値で分割する。TSに含まれる事例のなかで属性aの値がvである事例の集合をTSvとする。

ステップ6:全てのvについて、もしTSvが空集合でないなら、PDT(TSv)を実行し、TSvに対する決定木DTvを生成する*

$$\sum_{n\in\text{リーフノード}} \frac{m(n)}{M} \times \frac{e(n)+k-1}{m(n)+k} \qquad \text{(式3)}$$

nはリーフノードを表し、m(n)はノードnに属する訓練事例の数を表し、e(n)はノードnに属する訓練事例の中でnのクラスと一致していない(決定木によって正しく分類されない)事例の数を表し、kはクラスの数を表し、Mは総訓練事例数を表す。このラプラスエラーレートは決定木の誤り率を悲観的に評価する方法であって、値が小さい方が良い決定木である。

【００６２】・ステップ3-2ではID3を用いて、DTbvを生成したが、他の決定木生成アルゴリズムでもよい。

・ステップ4での決定木の評価は、ラプラスエラーレート以外の方法は、例えば決定木の大きさ(小さいほうが良い)などを用いてもよい。

【００６３】次に、具体的に説明する。

PDT({2,3,4,5,6,8})

1:事例2、8はクラスが負、3、4、5、6は正であるのでステップ1はパス。

【００６４】2:TS中に含まれている事例のうち、事例3と事例6とでは属性Xの値が異なるので、XはTSを分割可能である。同様に、属性Y,ZもTSを分割可能であるので、A={X,Y,Z}となる。

【００６５】3:Aに含まれる属性、すなわちX,Y,Zについて、決定木DTx,DTy,DTzを生成する。

DTxの生成

3-1:訓練事例集合TS={2,3,4,5,6,8}を、属性Xの値によって分割する。

【００６６】

TSx=0 ={2,3,4}(属性Xの値が0となる訓練事例の集合)

TSx=1 ={5,6,8}(属性Xの値が1となる訓練事例の集合)

3-2:TSx=0、TSx=1に対する決定木DTx=０、DTx=1を、ID3({2,3,4})、ID3({5,6,8})よって生成する。ID3の実行結果は図2の(a)と図2の(b)に示す。

【００６７】3-3:DTxを次のようにして、生成する。まずDTxのルートノードとして、属性Xをテスト属性とするテストノードを生成する。次に、この決定ノードの下のX=0に対応する部分決定木をDTx=0、X=1に対応する部分決定木をDTx=1とする。生成された決定木DTxを図2の(c)に示す。

【００６８】DTyの生成

3-1:訓練事例集合TS={2,3,4,5,6,8}を、属性Yの値によって分割する。

TSy=0 ={2,5,6}(属性Yの値が0となる訓練事例の集合)

TSy=1 ={3,4,8}(属性Yの値が1となる訓練事例の集合)

3-2:TSy=0、TSy=1に対する決定木DTy=0、DTy=1を、ID3({2,5,6})、ID3({3,4,8})よって生成する。ID3の実行結果は図3の(a)と図3の(b)に示す。

3-3:DTyを次のようにして、生成する。まずDTyのルートノードとして、属性Yをテスト属性とするテストノードを生成する。次に、この決定ノードの下のY=0に対応する部分決定木をDTy=0、Y=1に対応する部分決定木をDTy=1とする。生成された決定木DTyを図3の(c)に示す。

【００６９】DTzの生成

3-1:訓練事例集合TS={2,3,4,5,6,8}を、属性Zの値によって分割する。

TSz=0 ={3,5}(属性Zの値が0となる訓練事例の集合)

TSz=1 ={2,4,6,8}(属性Zの値が1となる訓練事例の集合)

3-2:TSz=0、TSz=1に対する決定木DTz=0、DTz=1を、ID3({3,5})、ID3({2,4,6,8})よって生成する。ID3の実行結果は図4の(a)と図4の(b)に示す。

【００７０】3-3:DTzを次のようにして、生成する。まずDTzのルートノードとして、属性Zをテスト属性とするテストノードを生成する。次に、この決定ノードの下のZ=0に対応する部分決定木をDTz=0、Z=1に対応する部分決定木をDTz=1とする。生成された決定木DTzを図4の(c)に示す。

【００７１】4:Aに含まれる属性、すなわちX,Y,Zについて、決定木DTx,DTy,DTzのラプラスエラーレートを計算すると、下の表のようになる。DTx、DTyがラプラスエラーレートを最小とするので、最適な属性としては、属性XまたはYのどちらかをランダムに選択される。ここではXが選択されたものとする。

【００７２】

決定木	ラプラスエラーレート
DTx	0.277778
DTy	0.277778
DTz	0.305556

5:訓練事例集合TS={2,3,4,5,6,8}を選択きれた属性Xの値で分割する。

【００７３】

TS0={2,3,4} (属性Xの値が0とな訓練事例の集合)

TS1={5,6,8} (属性Xの値が1とな訓練事例の集合)

6:TS0、TS1に対する決定木DT0、DT1を、それぞれPDT({2,3,4})、PDT({5,6,8})を実行することにより生成する。

【００７４】DT0 = PDT({2,3,4})

1:事例2は負、事例3、4は正であるので、ステップ1はパス。

2:事例2、3、4は全て属性Xの値が0であるので、Xの値では{2,3,4}を分割できない。事例2と事例3では属性Yの値が異なるので、属性Yでは分割可能。

【００７５】3:Aに含まれる属性、すなわちY,Zについて、決定木DTy,DTzを生成する。

DTyの生成

3-1:訓練事例集合TS={2,3,4}を、属性Yの値によって分割する。

【００７６】

TSy=0 ={2}(属性Yの値が0となる訓練事例の集合)

TSy=1 ={3,4}(属性Yの値が1となる訓練事例の集合)

3-2:TSy=0、TSy=1に対する決定木DTy=o、DTy=1を、ID3({2})、ID3({3,4})よって生成する。ID3の実行結果は図5の(a)と図5の(b)に示す。

【００７７】3-3:DTyを次のようにして、生成する。まずDTyのルートノードとして、属性Yをテスト属性とするテストノードを生成する。次に、この決定ノードの下のY=0に対応する部分決定木をDTy=0、Y=1に対応する部分決定木をDTy=1とする。生成された決定木DTyを図5の(c)に示す。

【００７８】DTzの生成

3-1:訓練事例集合TS={2,3,4}を、属性Zの値によって分割する。

TSz=0={3}(属性Zの値が0となる訓練事例の集合)

TSz=1 ={2,4}(属性Zの値が1となる訓練事例の集合)

3-2:TSz=0、TSz=1に対する決定木DTz=0、DTz=1を、ID3({3})、ID3({2,4})よって生成する。ID3の実行結果は図6の(a)と図6の(b)に示す。

【００７９】3-3:DTzを次のようにして、生成する。まずDTzのルートノードとして、属性Zをテスト属性とするテストノードを生成する。次に、この決定ノードの下のZ=0に対応する部分決定木をDTz=0、Z=1に対応する部分決定木をDTz=1とする。生成された決定木DTzを図6の(c)に示す。

【００８０】4:Aに含まれる属性、すなわちY,Zについて、決定木DTy,DTzのラプラスエラーレートを計算すると、下の表のようになる。DTyがラプラスエラーレートを最小とするので、最適な属性aとしては属性Yが選択される。

【００８１】

決定木	ラプラスエラーレート
DTy	0.277778
DTz	0.333333

5:訓練事例集合TS={2,3,4}を選択された属性Yの値で分割する。

【００８２】

TS0={2}(属性Yの値が0となる訓練事例の集合)

TS1={3,4}(属性Yの値が1となる訓練事例の集合)

6:TS0、TS1に対する決定本DT0,DT1を、それぞれPDT({2})、PDT({3,4})を実行することにより生成する。

【００８３】DT0 = PDT({2})

1:事例2のクラスは負であるので、クラス負のリーフノードを生成し、それを出力して、PDT({2})を終了する。

【００８４】DT1 = PDT({3,4})

1:事例3,4のクラスはどちらも正であるので、クラス正のリーフノードを生成し、それを出力して、PDT({3,4})を終了する。

【００８５】7:テスト属性がYであって、Y=0に相当する子ノードがDT0のルートノード、Y=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。この決定ノードがルートノードになる決定木を出力し、PDT({2,3,4})を終了。

【００８６】DT1=PDT({5,6,8})

1:事例5,6は正、事例8は負であるので、ステップ1はパス。

2:事例2,3,4は全て属性Xの値が1であるので、Xの値では{5,6,8}を分割できない。事例5と事例8では属性Yの値が異なるので、属性Yでは分割可能。同様に属性Zでも分割可能。よってA={Y,Z}となる。

【００８７】3:Aに含まれる属性、即ちY,Zについて、決定木DTy、DTzを生成する。

DTyの生成

3-1:訓練事例集合TS={5,6,8}を、属性Yの値によって分割する。

【００８８】

TSy=0 ={5,6}(属性Yの値が0となる訓練事例の集合)

TSy=1 ={8}(属性Yの値が1となる訓練事例の集合)

3-2:TSy=0、TSy=1に対する決定木DTy=0、DTy=1を、ID3({5,6})、ID3({8})よって生成する。ID3の実行結果は図6の(a)と図6の(b)に示す。

【００８９】3-3:DTyを次のようにして、生成する。まずDTyのルートノードとして、属性Yをテスト属性とするテストノードを生成する。次に、この決定ノードの下のY=0に対応する部分決定木をDTy=0、Y=1に対応する部分決定木をDTy=1とする。生成された決定木DTyを図6の(c)に示す。

【００９０】DTzの生成

3-1:訓練事例集合TS={5,6,8}を、属性Zの値によって分割する。

TSz=0 ={5}(属性Zの値が0となる訓練事例の集合)

TSz=1 ={6,8}(属性Zの値が1となる訓練事例の集合)

3-2:TSz=0、TSz=1に対する決定木DTz=0、DTz=1を、ID3({5})、ID3({6,8})よって生成する。ID3の実行結果は図7の(a)と図7の(b)に示す。

13

【００９１】3-3:DTzを次のようにして、生成する。まずDTzのルートノードとして、属性Zをテスト属性とするテストノードを生成する。次に、この決定ノードの下のZ=0に対応する部分決定木をDTz=0、Z=1に対応する部分決定木をDTz=1とする。生成された決定木DTzを図7の(c)に示す。

【００９２】4:Aに含まれる属性、すなわちY,Zについて、決定木DTy,DTzのラプラスエラーレートを計算すると、下の表のようになる。DTyがラプラスエラーレートを最小とするので、最適な属性としては属性Yが選択される。

【００９３】

決定木	ラプラスエラーレート
DTy	0.277778
DTz	0.333333

5:訓練事例集合Ts={5,6,8}を選択された属性Yの値で分割する。

【００９４】

TS0={5,6}(属性Yの値が0となる訓練事例の集合)

TS1={8}(属性Yの値が1となる訓練事例の集合)

6:TS0、TS1に対する決定本DT0,DT1を、それぞれPDT({5,6})、PDT({8})を実行することにより生成する。

【００９５】PDT({5,6})

1:事例5,6のクラスはどちらも正であるので、クラス正のリーフノードを生成し、それを出力して、PDT({5,6})を終了する。

【００９６】PDT({8})

1:事例8のクラスは負であるので、クラス負のリーフノードを生成し、それを出力して、PDT({8})を終了する。

【００９７】7:テスト属性がYであって、Y=0に相当する子ノードがDT0のルートノード、Y=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。この決定ノードがルートノードになる決定木を出力し、PDT({5,6,8})を終了。

【００９８】7:テスト属性がXであって、X=0に相当する*

14

*子ノードがDT0のルートノード、X=1に相当する子ノードがDT1のルートノードである決定ノードを生成する。この決定ノードがルートノードになる決定木を出力し、PDT({2,3,4,5,6,8})を終了。

【００９９】以上により生成した決定木を図8に示す。従来方式(ID3)で生成した図11の決定木とは異なり、本発明の決定木では属性Xと属性Yの排他的論理和を表す決定木を正しく求めることができた。実際、訓練事例集合に含まれず、ID3で生成した決定木(図11)では誤って分類された事例1、事例7も、本発明では正しく分類されている。このように、本発明によれば、正しい図8の決定木を自動生成することが可能となる。

【０１００】また、上述した決定木では、テスト属性およびその値によって分岐するので、場合によっては必要のない分岐を行う可能性がある。例えば０、１、２、３という４個の値をとる３個の属性Ｘ、Ｙ、Ｚが存在し、学習したいクラスが（Ｘ＝０ｏｒＸ＝１）ａｎｄ（Ｙ＝０ｏｒＹ＝１）であるとき、テスト属性としてＸあるいはＹが選ばれることが望ましい。しかし、Ｘが選ばれても、Ｘ＝０となる訓練事例とＸ＝１となる訓練事例が分割されてしまうこととなる。これでは、決定ノードの下の各部分木を生成するための事例の数が少なくなってしまい、その結果、通常得られる決定木の精度が小さくなるという欠点を持つ。このため、仮の属性を生成し、それらのなかからテスト属性を選択することによって、この欠点を解決する。即ち、仮の属性は、事例の学習したいクラスの結果をほぼ同じくする属性の値を１つにまとめる（例えば上記例ではＸが０、１、２、３の４つの値を持つときに０、１を１つにまとめる）ように生成する。これにより、上記例では、Ｘ＝０と、Ｘ＝１とを分割した決定木を生成するという無駄な分割を回避することが可能となる。これを数式で表現すると下記のようになる。

【０１０１】・例えば属性Ｘから以下のように仮の属性Ｘ’やＸ’’を生成する。

```
X'＝0　if and only if X＝0 or X＝1
X'＝1　otherwise
X''＝0　if and only if X＝0
X''＝1　if and only if X＝1，2
X''＝2　if and only if X＝3
```

また、上述した決定木では、ある１つの属性から生成される仮の属性の数は、もとの属性のとり得る値の数が大きい場合、非常に多くなる。例えばもとの属性のとりうるある値の数が２のときは、仮の属性の数は１個（もとの属性と等価な属性）であり、３のときは４個であり、４のときは１３個であり、５のときは５１個である。このように属性の取り得る値の数が増えると急激に数が増え、これら全てについて最適な決定木を決めるために既述した評価を行うのは、非常に時間がかかる。このため、いわゆる山登り法を下記のようにして用いる。

【０１０２】Ｘ’：｛０，１｝、｛２｝、｛３｝というノーテーションで、値として０、１、２、３をとる属性Ｘからつくられる仮の属性で、その値が３つの値をとり、もしもとの属性の値が０と１ならば（３つの値のうち）最初の値、２ならば２番目の値、３ならば３番目の値をとるような属性を表すものとする。値として、０、１、２、３をとる属性Ｘに対して、以下説明する。

【０１０３】もともとの属性Ｘと等価な属性

Ｘ４，１：｛０｝、｛１｝、｛２｝、｛３｝

から開始する。次に、Ｘ４，１の４つの値のうち、２つ

15

を同一とみなすことによって生成される属性
Ｘ３，１：｛０，１｝、｛２｝、｛３｝
Ｘ３，２：｛０，２｝、｛１｝、｛３｝
Ｘ３，３：｛０，３｝、｛１｝、｛２｝
Ｘ３，４：｛０｝、｛１，２｝、｛３｝
Ｘ３，５：｛０｝、｛１，３｝、｛２｝
Ｘ３，６：｛０｝、｛１｝、｛２，３｝
を評価し、それらのなかで最も良いものを選択する。ここでは、Ｘ３，４が選択されたとする。次に、同様にして、Ｘ３，４をもとに、それらの値の２つを同一とみなすことによって生成される属性
Ｘ２，１：｛０，１，２｝、｛３｝
Ｘ２，２：｛０，３｝、｛１，２｝
Ｘ２，３：｛０｝、｛１，２，３｝
を評価する。これ以上値のマージを行えないので、仮の属性のマージは終了する。そして、これまでに生成されたなかから評価値の最良の属性を選択する。この例では、１３個の可能な属性のうち、１０個のみを評価して最良のものを選択している。もともとの属性が５個の値をとり得る場合には、５１個のうち、２０個を生成して評価すればよく、少ない処理量にすることが可能となる。尚、もとの属性のとり得る値の数をＮとすると、生成／評価する仮の属性の数は、下式で表される。

【０１０４】

$$1+\sum_{n=N}^{3}(nC2)$$

【０１０５】

【発明の効果】以上説明したように、本発明によれば、事例集合から決定木を生成するときに分岐可能な決定木を評価して評価値の最も高い決定木を生成することを繰り返すと共に、分岐するときに不要な分割を回避および山登り法を採用する構成を採用しているため、少ない処理回数で分類精度の良好な決定木を自動生成できる。これらにより、

（１）　本発明では、属性のクラスとの間の隠れた関連*

16

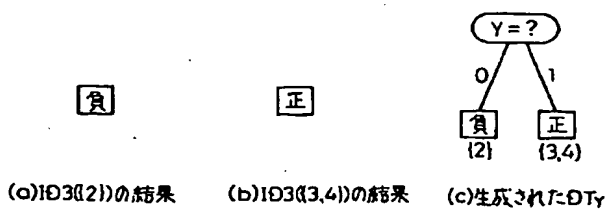
*性を、分割可能な属性によって分割した決定木を生成して評価値を求めて最良の決定木を選択することを繰り返することによって検出しているため、事例を正確に分類する決定木を自動生成することが可能となる。

【０１０６】（２）　決定木の分岐において、結果（クラス）が同じようになる属性の値をまとめて分岐しないようにして不要な分岐を回避して、正確な結果が得られる決定木を自動生成することが可能となる。

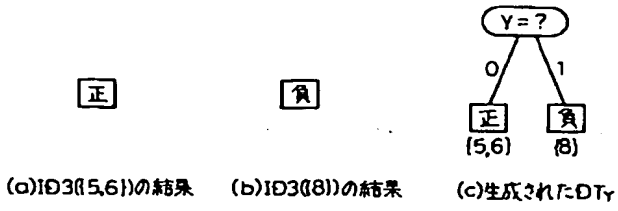
【０１０７】（３）　分割可能な属性の値で分岐する際に、属性が取り得る値の任意の２つを１つにまとめて評価を行って最も評価の高い属性を選ぶことを繰り返した後、全体の中で最も評価値の高い属性の値で分岐しているため、処理回数を削減して高速処理することが可能となる。

【図面の簡単な説明】
【図１】本発明のシステム構成図である。
【図２】本発明の具体例（その１）である。
【図３】本発明の具体例（その２）である。
【図４】本発明の具体例（その３）である。
【図５】本発明の具体例（その４）である。
【図６】本発明の具体例（その５）である。
【図７】本発明の具体例（その６）である。
【図８】本発明の具体例（その７）である。
【図９】事例である。
【図１０】決定木例である。
【図１１】従来方式により生成された決定木である。
【符号の説明】
１：データ入力部
２：決定木学習部
３：出力部
４：属性選択部
５：仮属性生成部
６：仮決定木生成部
７：決定木評価部
８：山登り部

【図５】

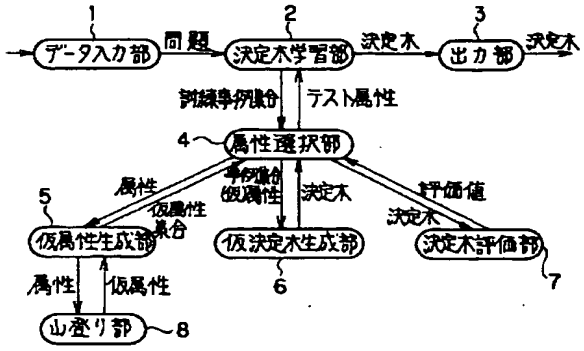


【図６】



【図１】

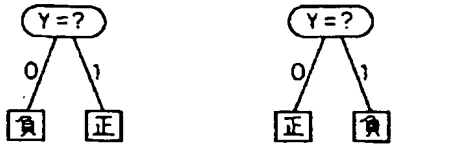
本発明のシステム構成図



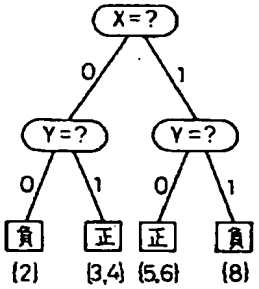
【図２】

本発明の具体例（その１）

(PDT({2,3,4,5,6,8}) のステップ３における DTx の生成)



(a)ID3({2,3,4}) の結果　(b)ID3({5,6,8}) の結果

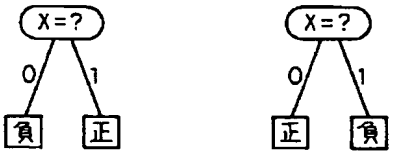


(c)生成された DTx

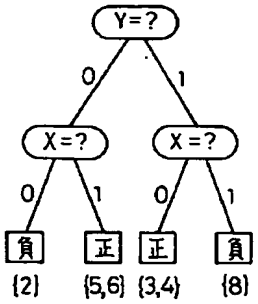
【図３】

本発明の具体例（その２）

(PDT({2,3,4,5,6,8}) のステップ３における DTy の生成)



(a)ID3({2,5,6}) の結果　(b)ID3({3,4,8}) の結果

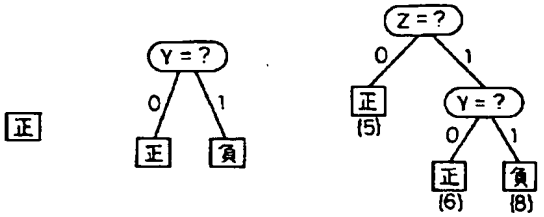


(c)生成された DTy

【図７】

本発明の具体例（その６）

(PDT({5,6,8}) のステップ３における DTz の生成)



(a)ID3({5}) の結果　(b)ID3({6,8}) の結果　(c)生成された DTz

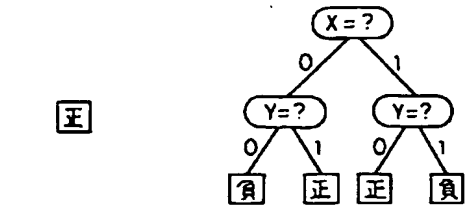
【図９】

事　例

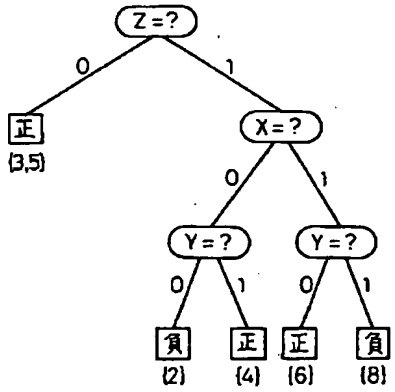
事例番号	属性値			クラス
	X	Y	Z	
1	0	0	0	負
2	0	0	1	負
3	0	1	0	正
4	0	1	1	正
5	1	0	0	正
6	1	0	1	正
7	1	1	0	負
8	1	1	1	負

【図４】

本発明の具体例(その3)
(PDT({2,3,4,5,6,8})のステップ3におけるDTzの生成)



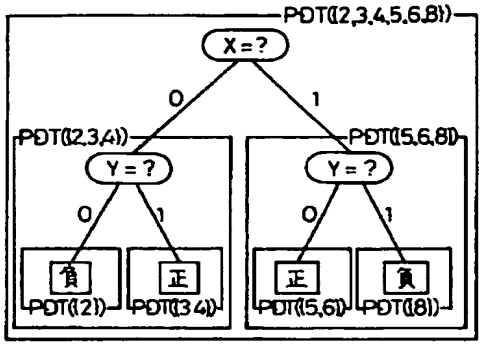
(a)ID3({3,5})の結果　　(b)ID3({2,4,6,8})の結果



(c)生成されたDTz

【図８】

本発明の具体例(その7)
(生成された決定木)



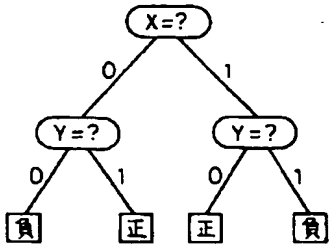
a=? はテスト属性がa(a=X or Y or Z)である決定ノードを表わす。

C はクラスがC(C=正 or 負)であるリーフノードを表わす

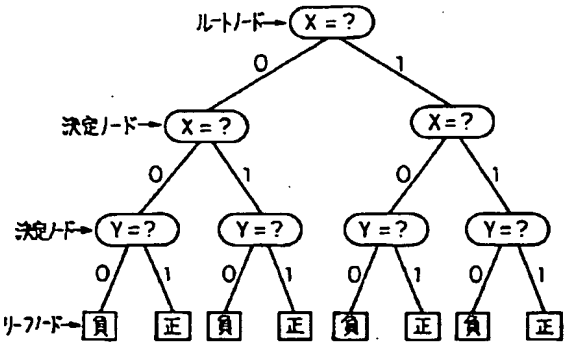
PDT(TS) はPDT({TS})によって生成される(部分)決定木を表わす。

【図１０】

決定木例



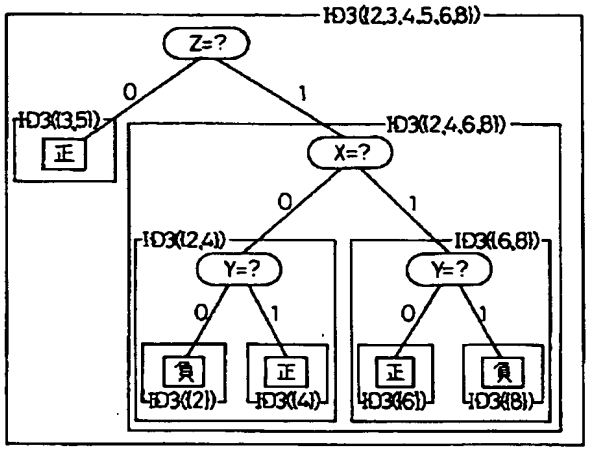
(a)属性Xの評価時に生成される決定木



(b)属性Zの評価時に生成される決定木

【図１１】

従来方式により生成された決定木



a=? はテスト属性がa(a=XorYorZ)である決定ノードを表わす。

C はクラスがC(C=正or負)であるリーフノードを表わす。

ID3(TS) はID3((TS))によって生成される(部分)決定木を表わす。